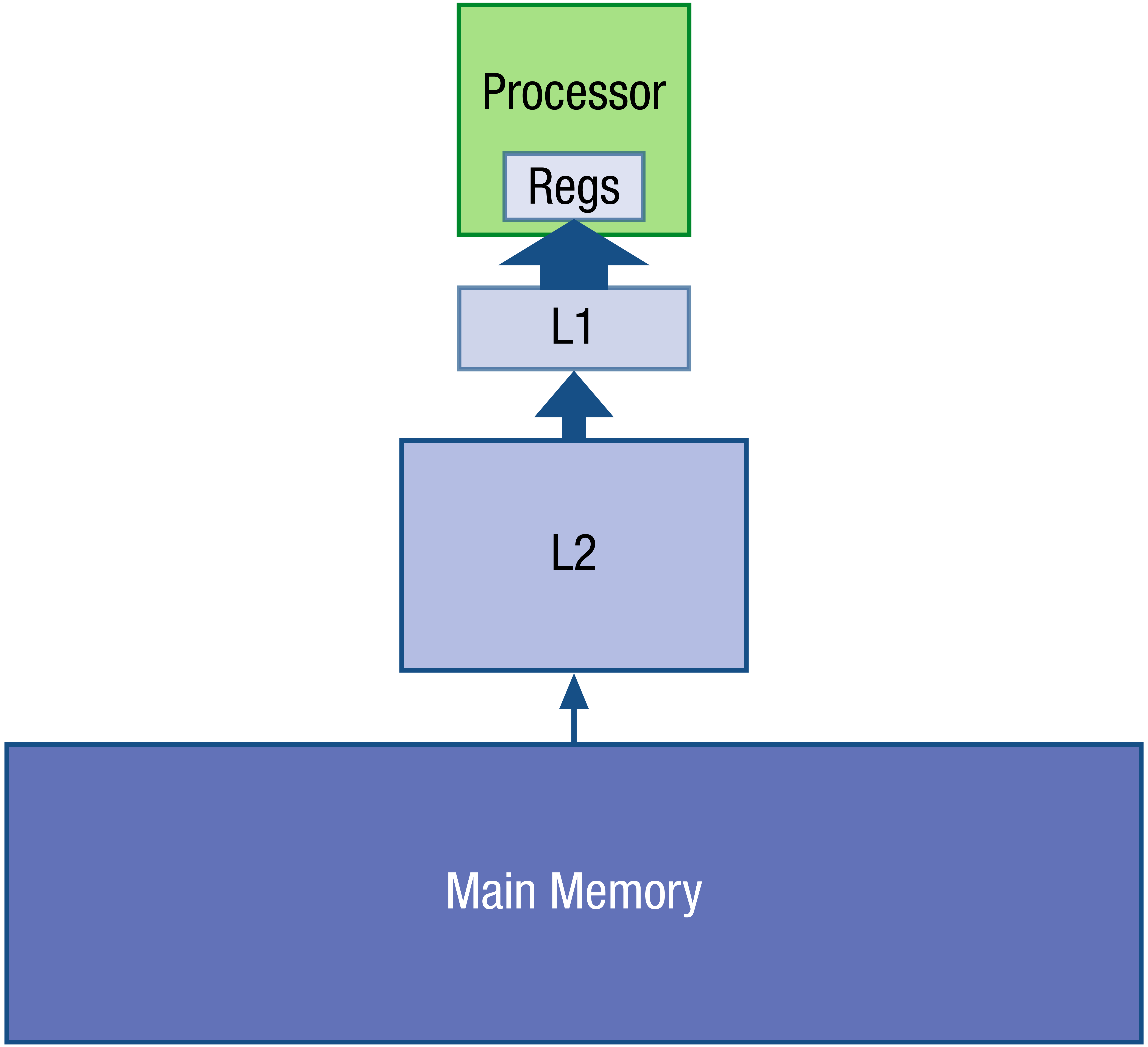
September 18, 2025

6.S894
Accelerated Computing

Lecture 3: Memory

Jonathan Ragan-Kelley

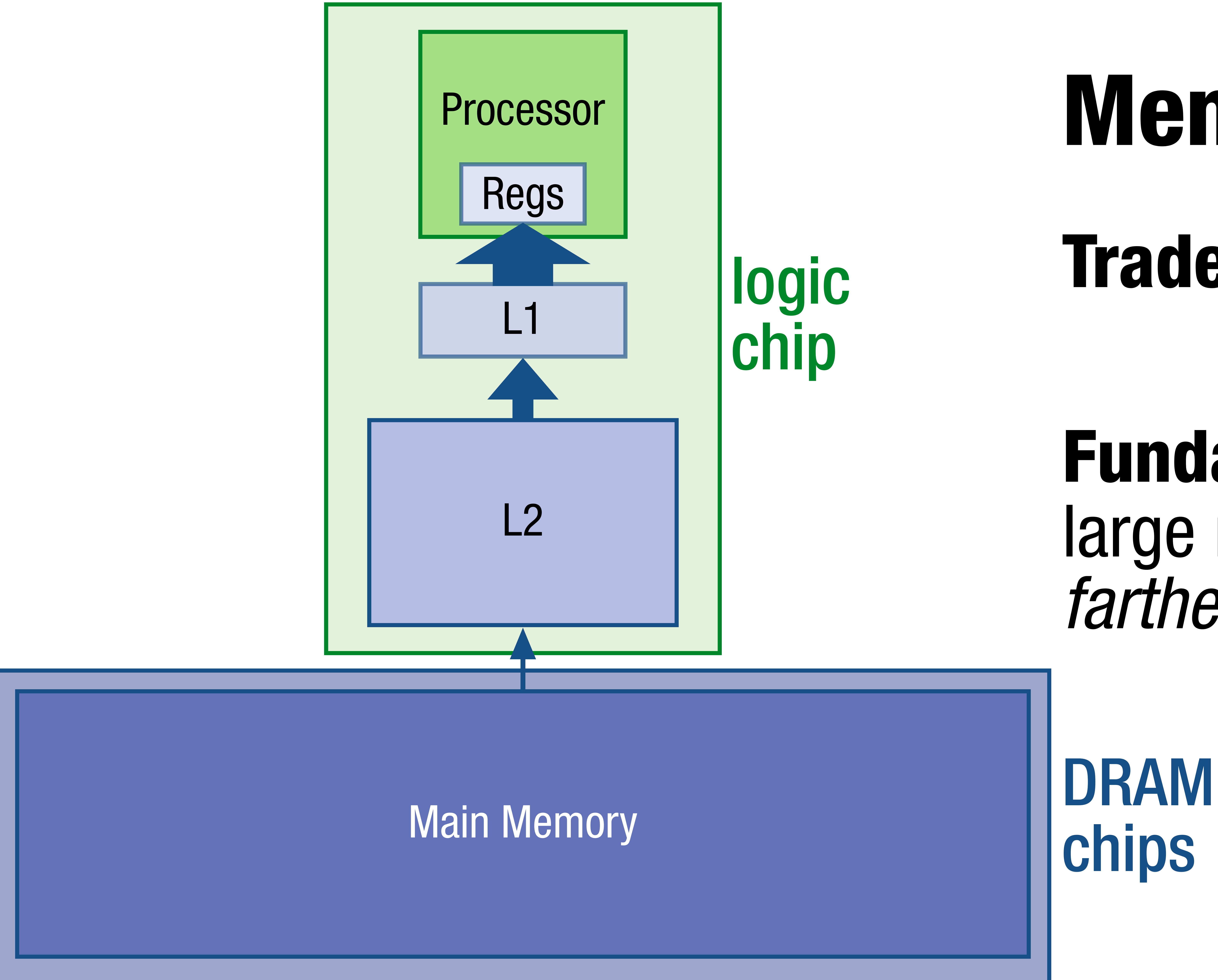
A Conventional **Memory Hierarchy**



Tradeoff: small, fast, close
vs. large, slow, far

Fundamental constraint:
large memories are
farther away, on average

A Conventional **Memory Hierarchy**



Tradeoff: small, fast, close
vs. large, slow, far

Fundamental constraint: large memories are *farther away, on average*

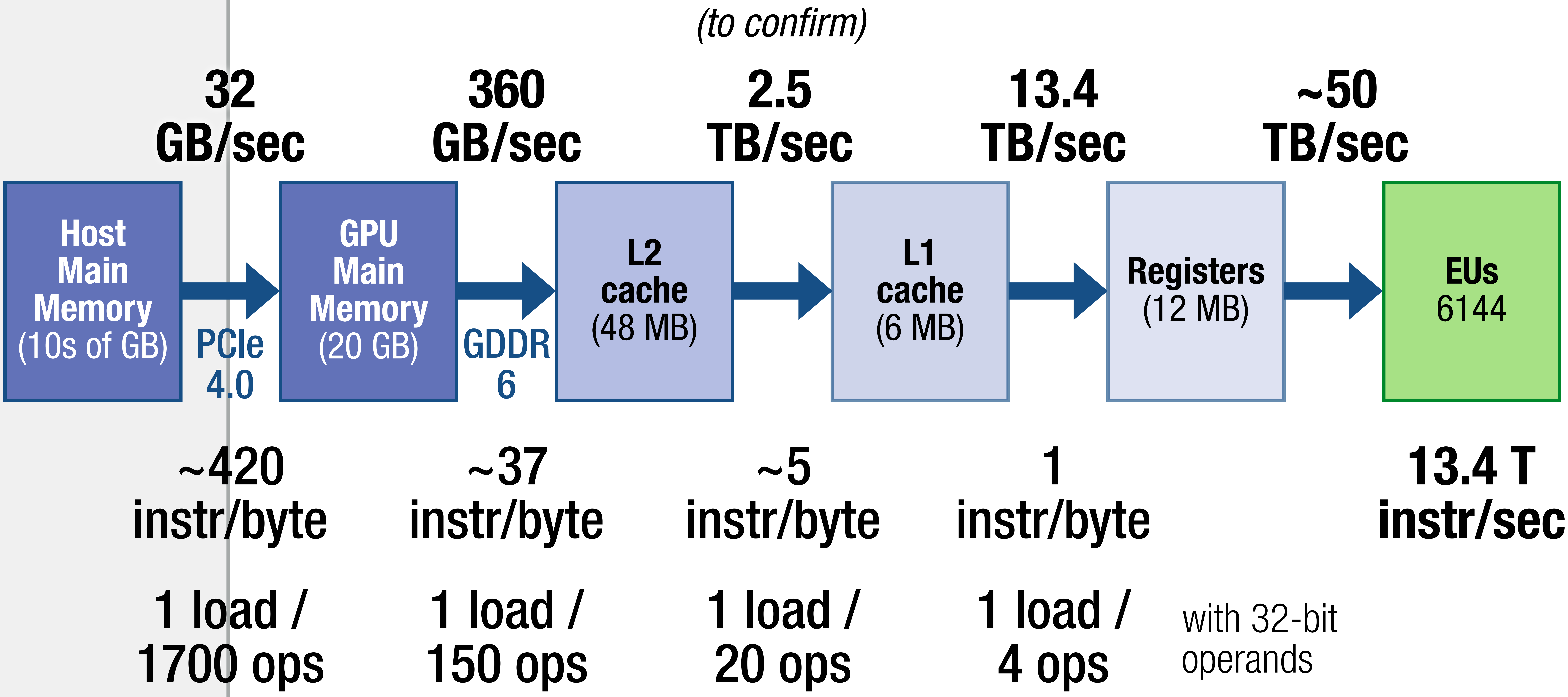
CPU
Processor
Regs
L1
L2
Main Memory
GPU
Processor
Regs
L1
L2
Main Memory

more parallelism

more registers

higher bandwidth, lower capacity

CPU
Processor
Regs
L1
L2
Main Memory
Host
PCI Express
Accelerator
GPU
Processor
Regs
L1
L2
Main Memory



(to confirm)

Host Main Memory (10s of GB)

32 GB/sec

PCIe 4.0

~420 instr/byte

1 load / 1700 ops

GPU Main Memory (20 GB)

360 GB/sec

GDDR 6

~37 instr/byte

1 load / 150 ops

L2 cache (48 MB)

2.5 TB/sec

~5 instr/byte

1 load / 20 ops

L1 cache (6 MB)

13.4 TB/sec

1 instr/byte

1 load / 4 ops

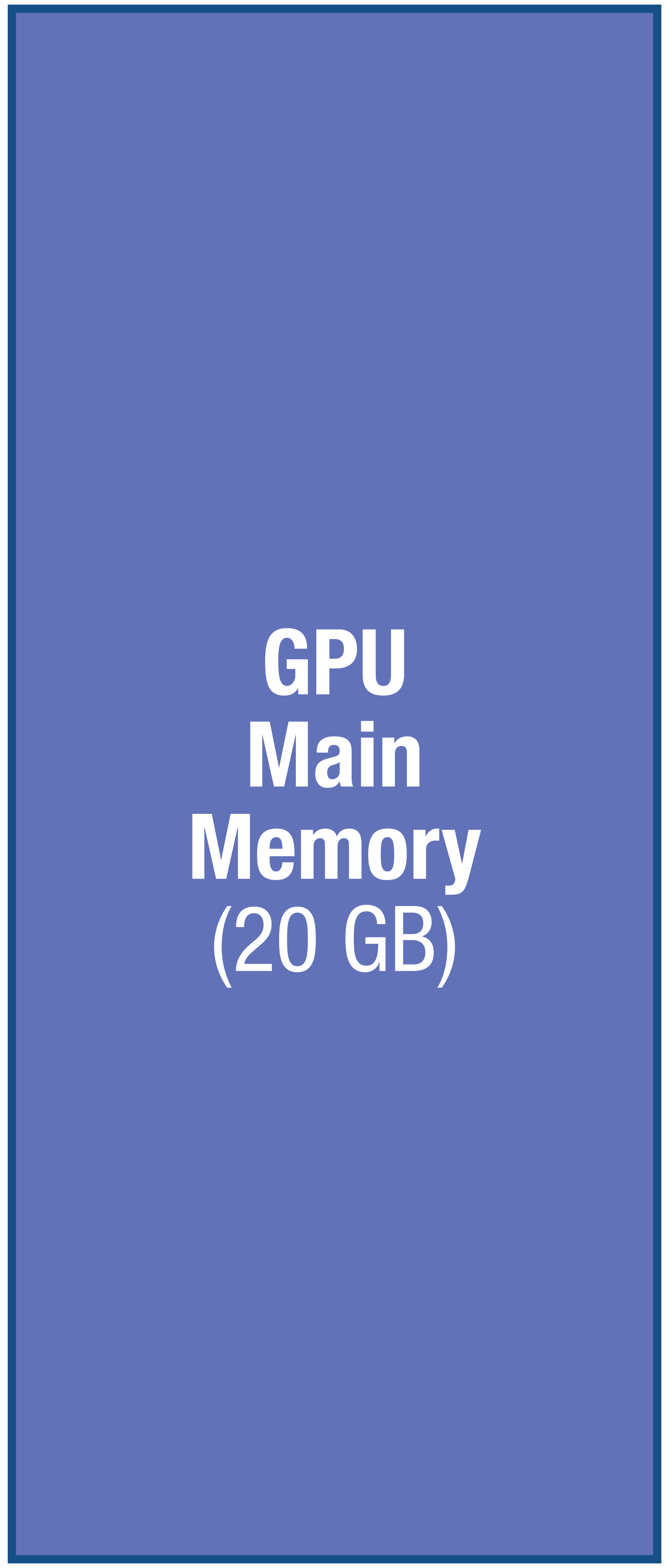
Registers (12 MB)

with 32-bit operands

~50 TB/sec

EUs 6144

13.4 T instr/sec



How can we get **more bandwidth?**

1. **Faster** bus frequency
2. **Wider** interface (parallelism)

Capacity: 1-4 GBytes x *n* chips

Interface: 32 bits

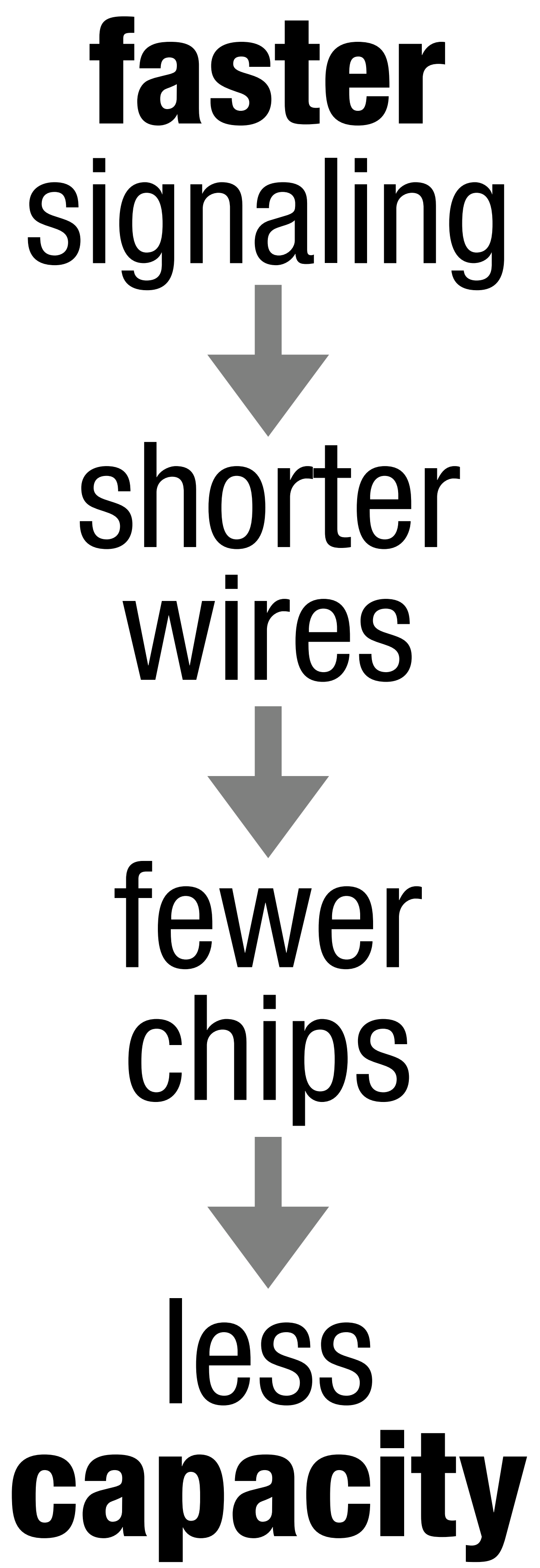
Speed: 6.4 GT/s

} 25.6 GB/sec

(DDR5-6400)

Increasing memory bus **frequency**

Tradeoff:



faster signaling

↓

shorter wires

↓

fewer chips

↓

less **capacity**

DRAM
GDDR6

Capacity: 4 GBytes **x *1* chips**

Interface: 32 bits

Speed: 18 GT/s

(GDDR6)

72 GB/sec

DRAM
DDR5

Capacity: 1-4 GBytes x *n* chips

Interface: 32 bits

Speed: 6.4 GT/s

(DDR5-6400)

25.6 GB/sec

Increasing memory bus **width** (parallelism!)

Aggregate: 5 chips

Capacity: 20 GBytes

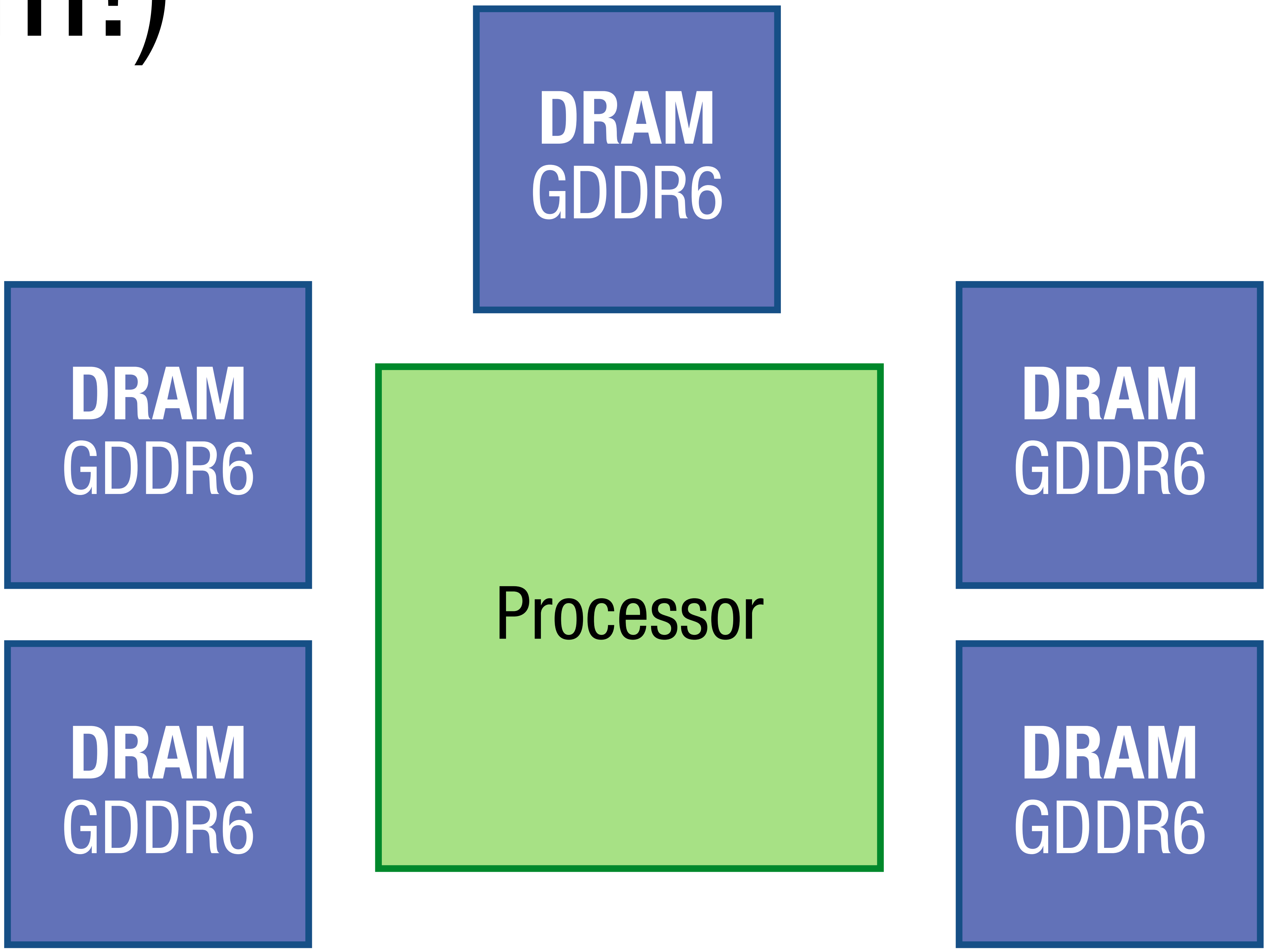
Interface: 160 bits

Speed: 18 GT/s
(GDDR6)

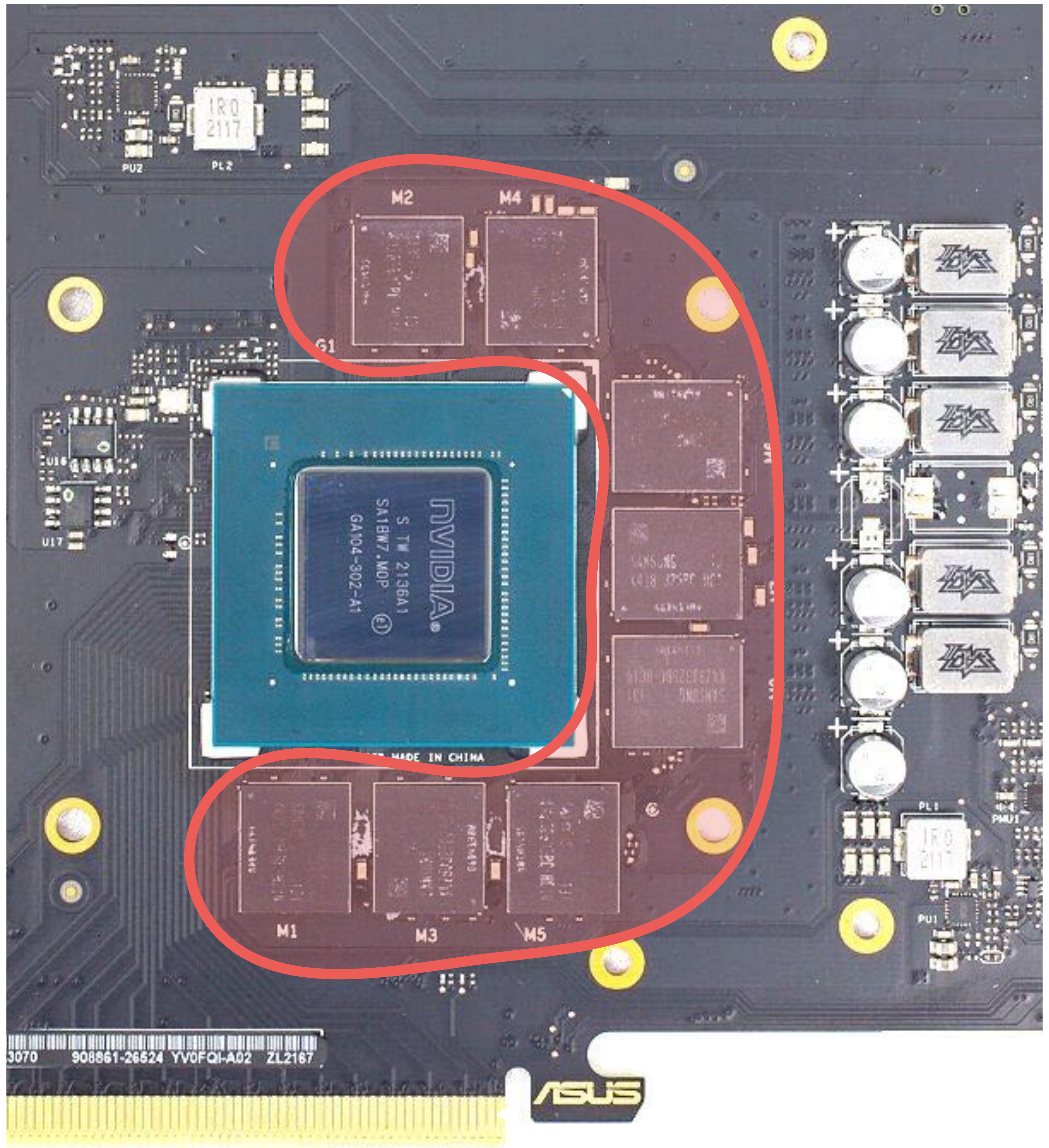
} 360 GB/sec

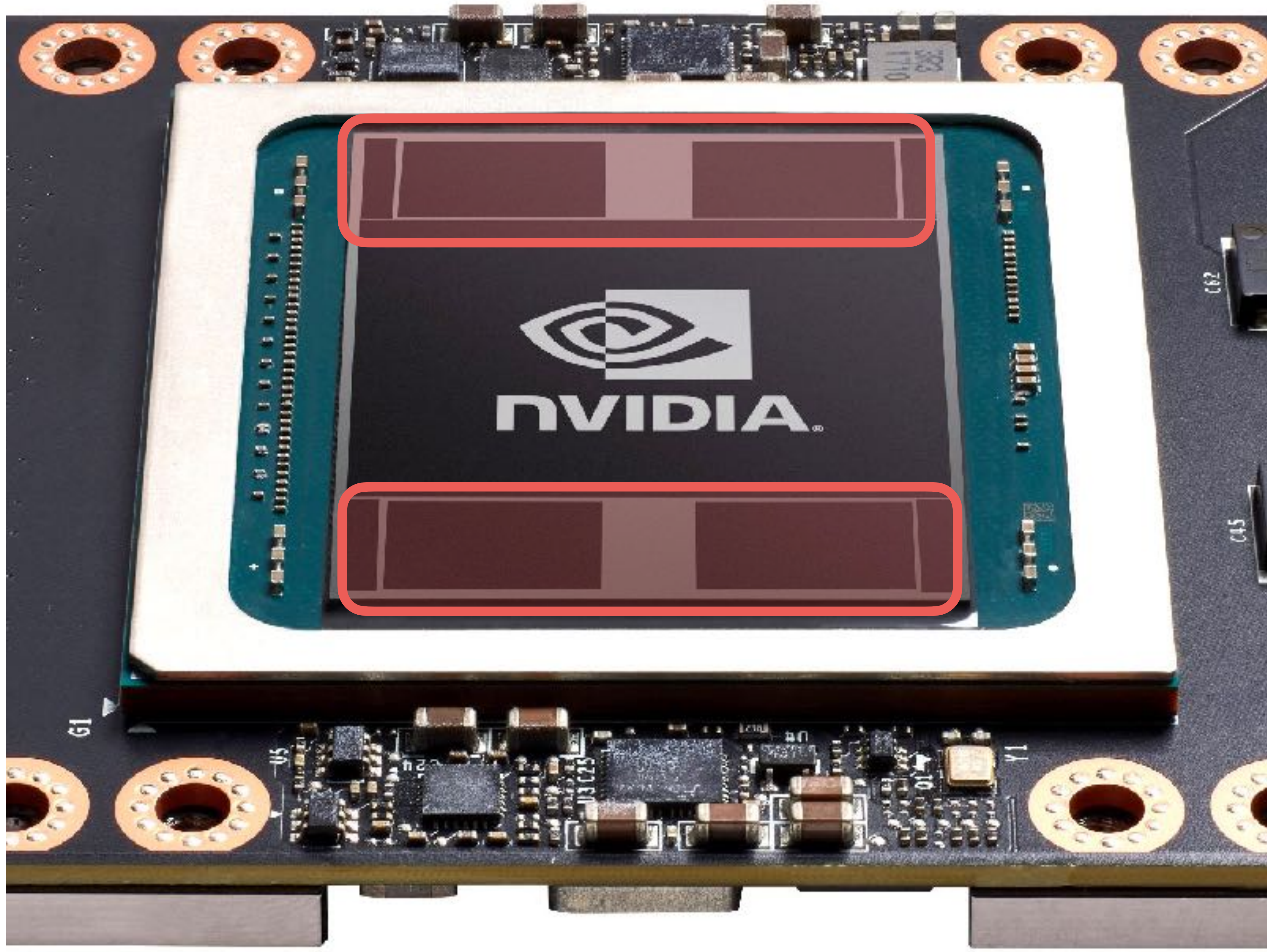
Limit: processor pins

Practical: ~384-512 bits
(12-16 chips)



Increasing memory bus width **further: package-level** integration





Increasing memory bus width **further:** **package-level** integration

Per-module:

Capacity: 24 GBytes

Interface: 2048 bits }

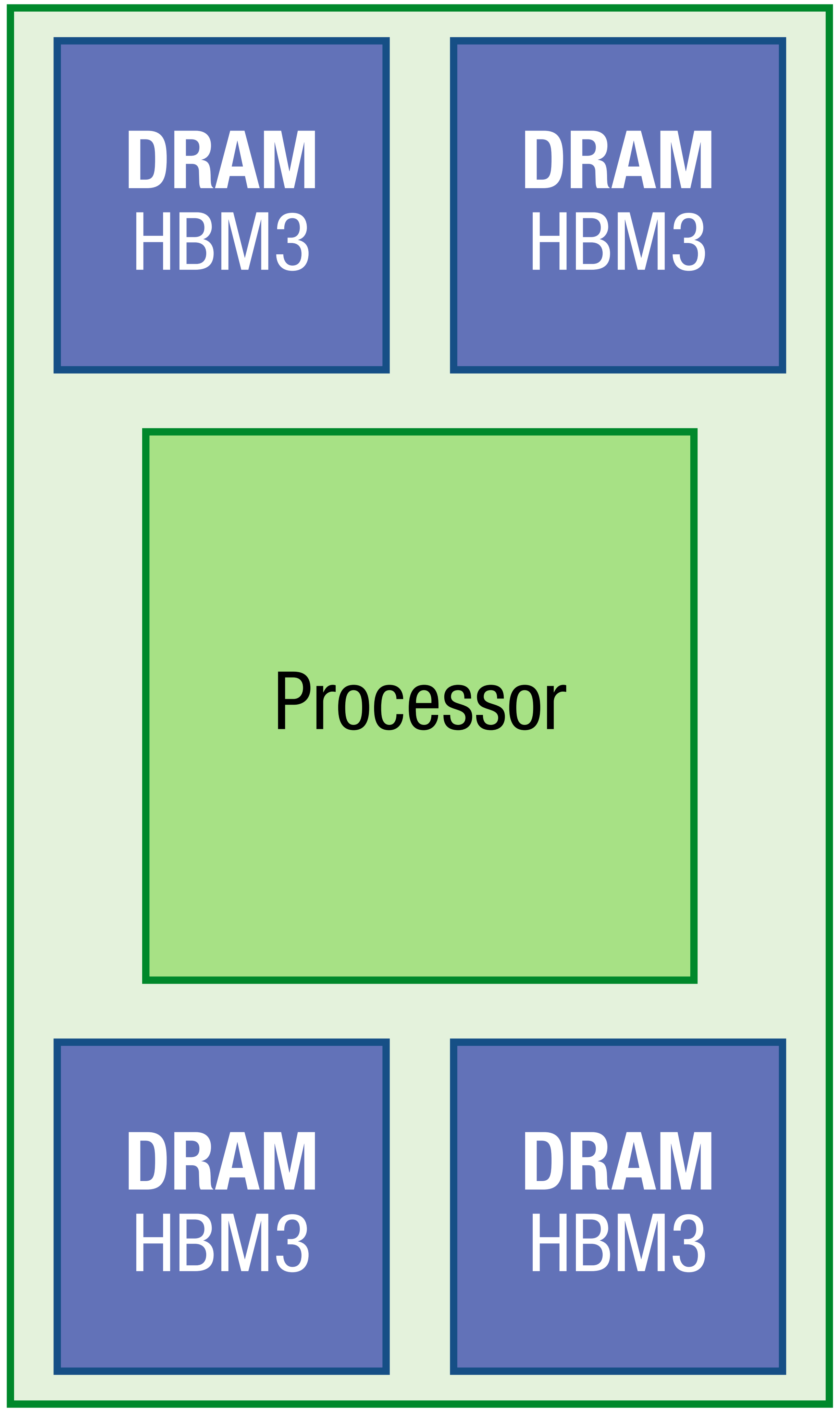
Speed: 3.2 GT/s } 819 GB/sec

(HBM3)

Aggregate:

Capacity: 96 GBytes

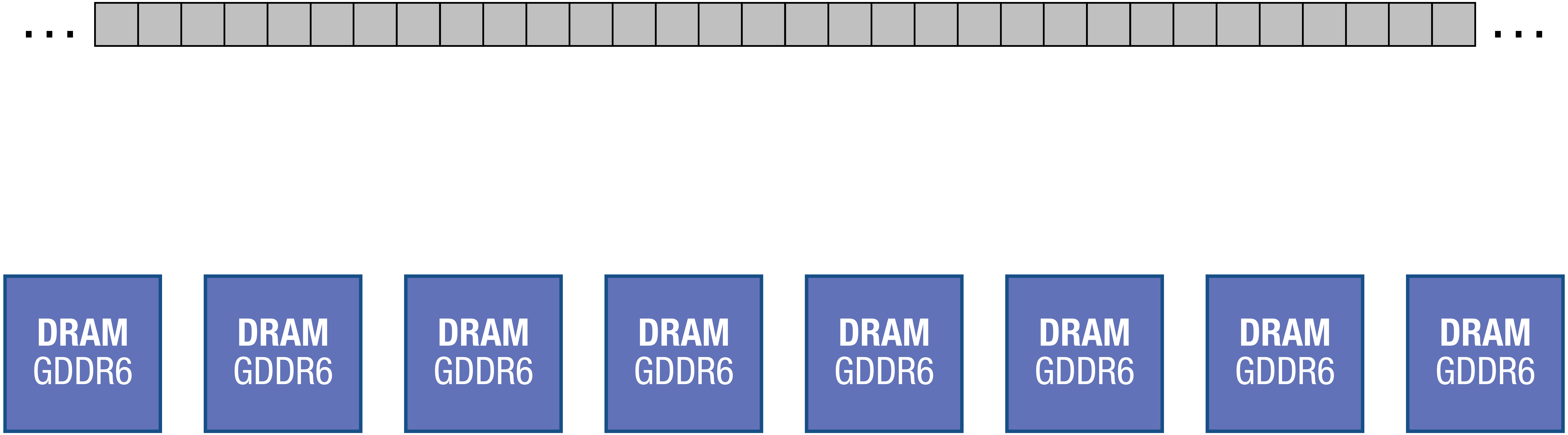
Interface: 8192 bits } 3.3 TB/sec



High bandwidth requires
wide memory interfaces

How can we
keep them fed?

Mapping **address space** to **memory channels**



DRAM GDDR6

DRAM GDDR6

DRAM GDDR6

DRAM GDDR6

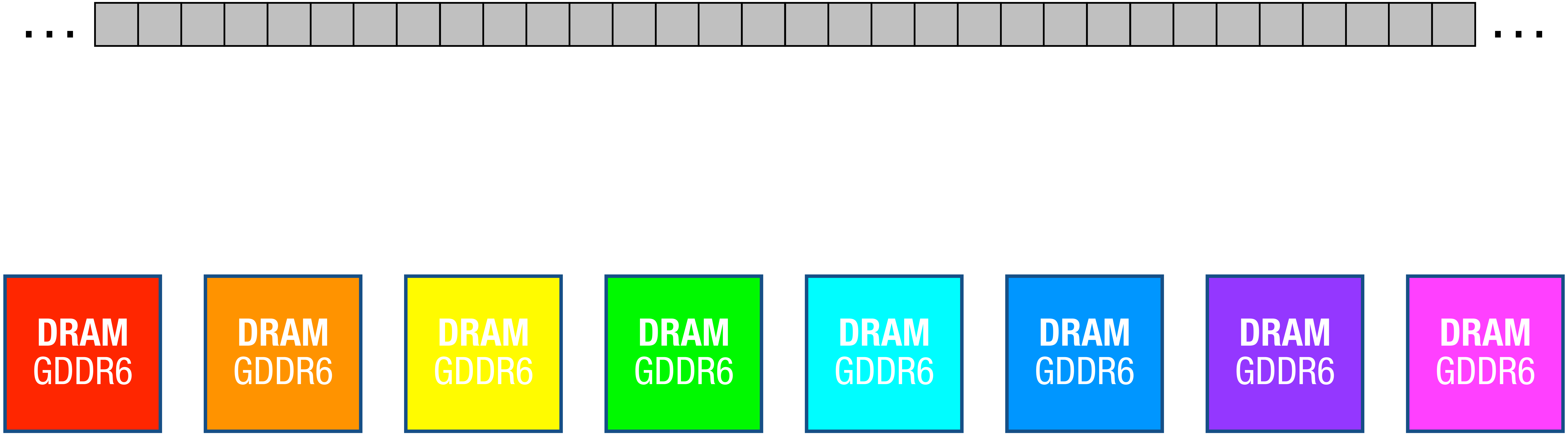
DRAM GDDR6

DRAM GDDR6

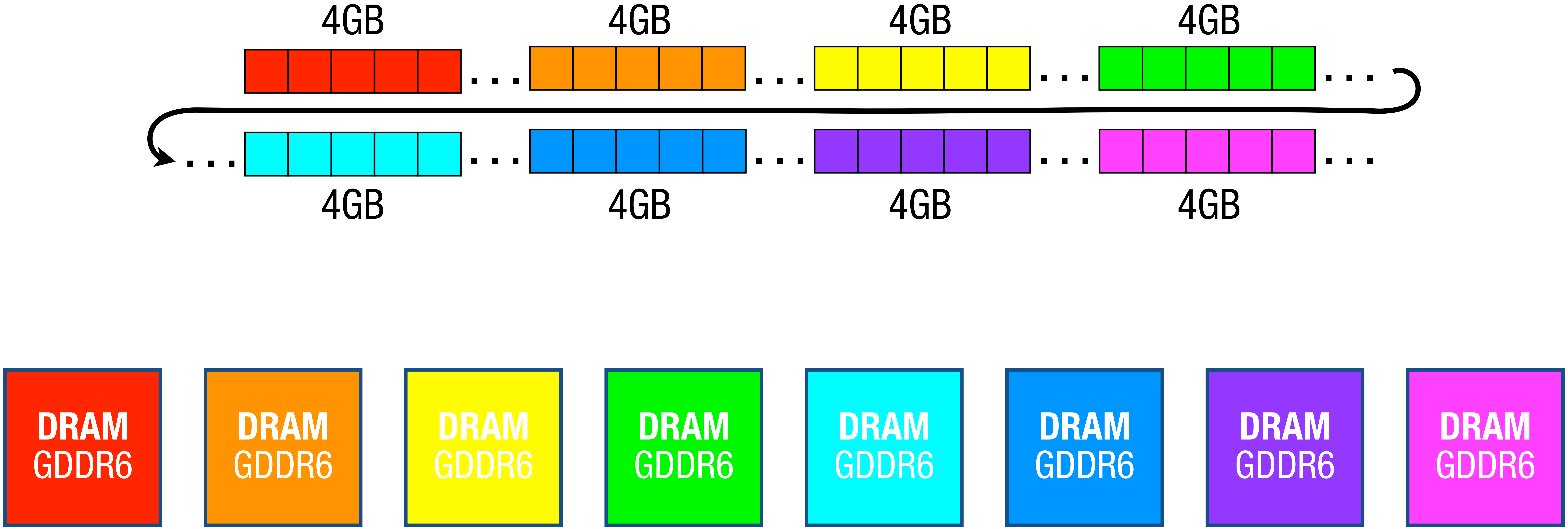
DRAM GDDR6

DRAM GDDR6

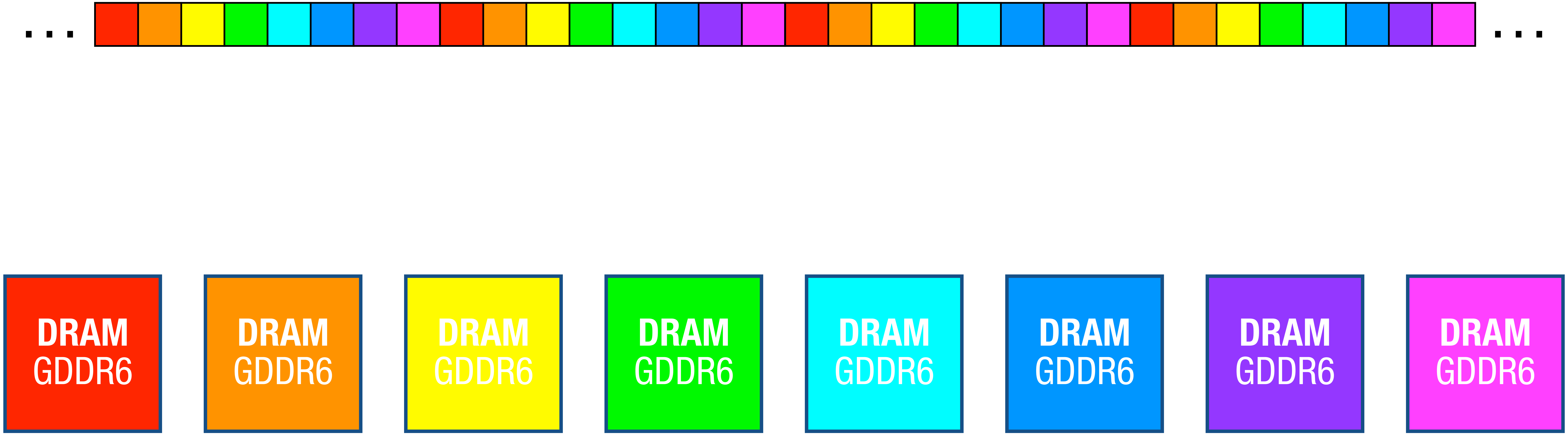
Mapping **address space** to **memory channels**



Mapping **address space** to **memory channels**



Mapping **address space** to **memory channels**



...

...

DRAM GDDR6

DRAM GDDR6

DRAM GDDR6

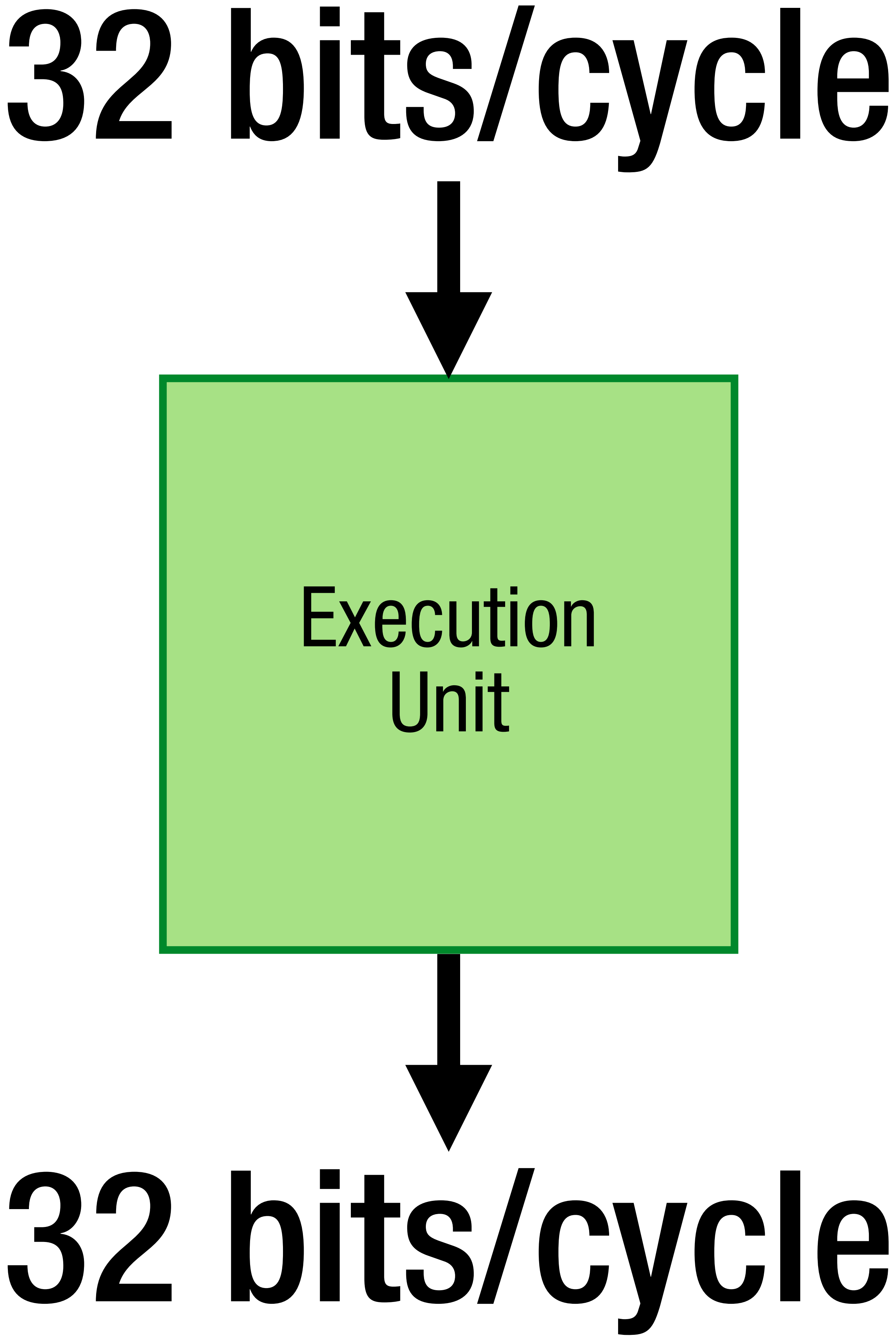
DRAM GDDR6

DRAM GDDR6

DRAM GDDR6

DRAM GDDR6

DRAM GDDR6



32 bits/cycle

Execution Unit

32 bits/cycle

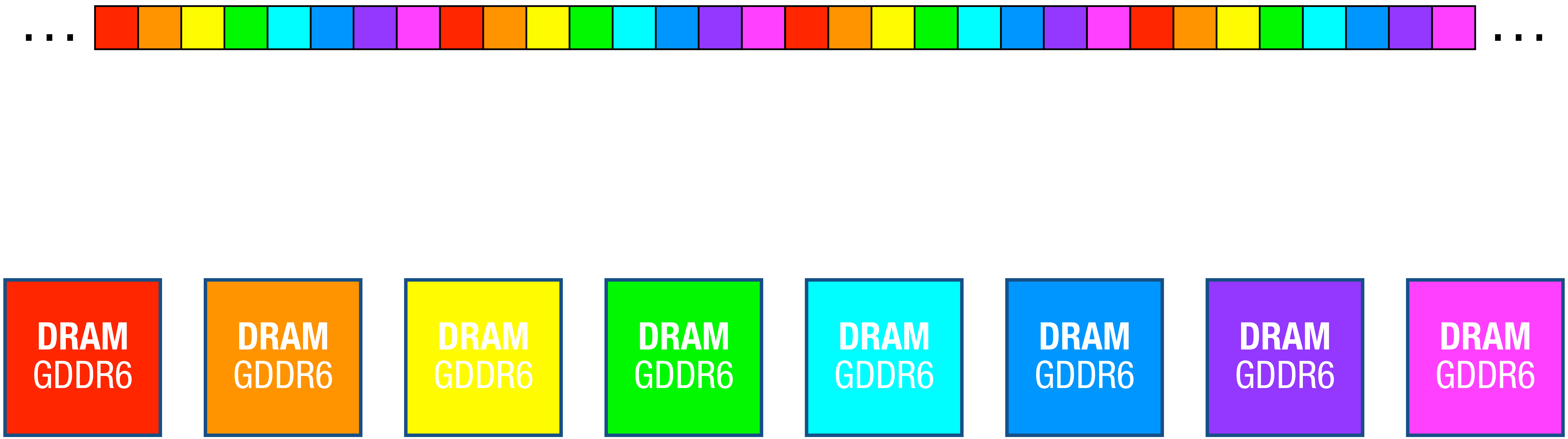
32 x 32 bits/cycle
…
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
EU
…
32 x 32 bits/cycle

1024 bits/cycle

···	i	i+1	i+2	i+3	i+4	i+5	i+6	i+7	i+8	i+9	i+10	i+11	i+12	i+13	i+14	i+15	i+16	i+17	i+18	i+19	···

1024 bits/cycle

Striping memory across channels generally gives **higher bandwidth**


. . .
. . .
DRAM GDDR6
DRAM GDDR6
DRAM GDDR6
DRAM GDDR6
DRAM GDDR6
DRAM GDDR6
DRAM GDDR6
DRAM GDDR6